


IoT Smart Water Management System

 *Yahya Al Benni – Oussama Khatib – Elias Jamhour*

Introduction

- IoT devices need efficient communication protocols for data exchange.
 - Two commonly used protocols for constrained devices: **MQTT** (Message Queuing Telemetry Transport) and **CoAP** (Constrained Application Protocol).
 - This presentation compares both and explains their **role in monitoring the main tank water level, house tank water level, turbidity sensor, user notifications, and pump control** in our project.
-

MQTT Overview

What is MQTT?

- A lightweight, **publish-subscribe** messaging protocol.
- Uses a **broker** to relay messages between devices.
- Designed for low-bandwidth, high-latency networks.

How it Works:

1. A device **publishes** data to a topic.
2. A broker **receives** the message and distributes it.
3. Other devices **subscribe** to the topic to receive updates.

Advantages:

- ✓ Low bandwidth usage.
 - ✓ Reliable message delivery (QoS levels).
 - ✓ Works over TCP/IP, ensuring stable communication.
-

CoAP Overview

What is CoAP?

- A **request-response** protocol similar to HTTP but optimized for IoT.
- Uses **UDP** instead of TCP for lightweight operation.
- Designed for constrained networks and low-power devices.

How it Works:

1. A client **sends** a request to a server (e.g., "GET water level").
2. The server **responds** with the requested data.
3. Supports **confirmable** and **non-confirmable** messages.

Advantages:

- ✓ Works efficiently with resource-constrained devices.
- ✓ Faster response time due to UDP.
- ✓ Supports multicast communication.

Comparison (MQTT vs. CoAP)

Feature	MQTT	CoAP
Communication	Publish-Subscribe	Request-Response
Transport Protocol	TCP	UDP
Reliability	High (QoS levels)	Lower (optional acknowledgments)
Bandwidth Usage	Low	Very Low
Latency	Moderate	Low
Best For	Real-time monitoring	Command-response IoT interactions

Implementation in Our Project

Both **MQTT** and **CoAP** were used to monitor the **main tank water level**, **house tank water level**, **turbidity sensor**, provide **user notifications**, and enable **pump control**.

MQTT:

- The **main tank and house tank sensors** publish real-time water levels.
- The **turbidity sensor** publishes water quality data.
- The **mobile app subscribes** to receive **real-time notifications** when:
 - **House tank level is low** (user is alerted to reduce water usage).
 - **Turbidity is high** (user is warned about water quality).
 - **Main tank is full** (pump stops automatically).

CoAP:

- The **pump controller requests** sensor data when needed.
- The **server responds** with:
 - **Main tank water level**
 - **House tank water level**

- Turbidity status
- The system **triggers pump control actions** based on sensor feedback.

Results and Observations

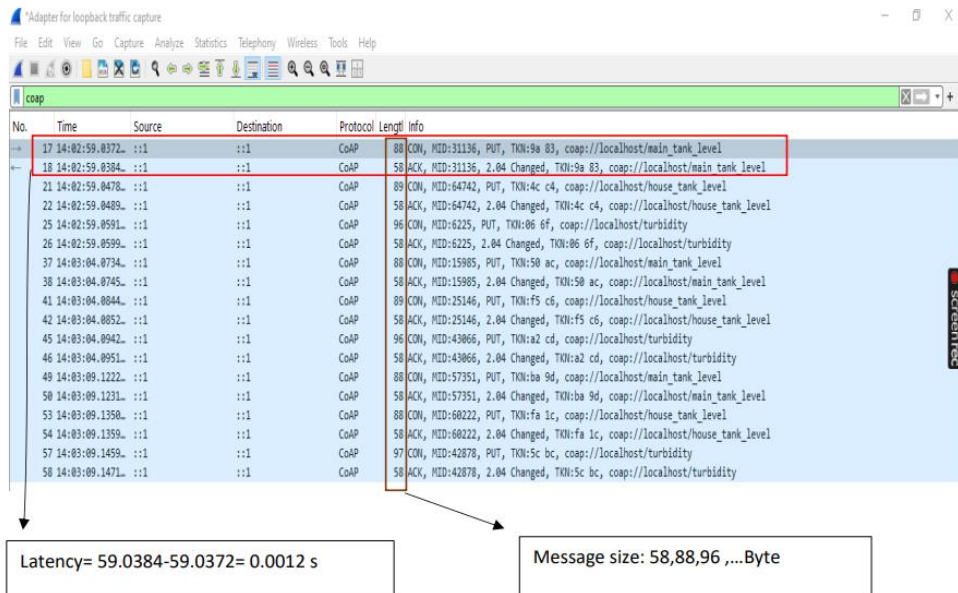
MQTT

No.	Time	Source	Destination	Protocol	Length	Info
10	12:07:05.4903..	192.168.1.109	137.135.83.217	MQTT	74	Connect Command
11	12:07:05.4908..	192.168.1.109	137.135.83.217	MQTT	68	Connect Ack
13	12:07:05.5312..	192.168.1.109	137.135.83.217	MQTT	82	Subscribe Request (id=1) [water_level/main_tank]
15	12:07:05.6521..	137.135.83.217	192.168.1.109	MQTT	132	Subscribe Request (id=2) [water_level/house_tank], Subscribe Request (id=3) [water_quality/turbidity], Subscribe...
17	12:07:05.8200..	137.135.83.217	192.168.1.109	MQTT	68	Subscribe Ack (id=1)
36	12:07:19.4176..	192.168.1.109	137.135.83.217	MQTT	68	Subscribe Ack (id=2), Subscribe Ack (id=3), Subscribe Ack (id=4)
38	12:07:19.6972..	137.135.83.217	192.168.1.109	MQTT	79	Connect Command
39	12:07:19.6978..	192.168.1.109	137.135.83.217	MQTT	68	Connect Ack
41	12:07:19.7342..	192.168.1.109	137.135.83.217	MQTT	79	Subscribe Request (id=1) [user/notifications]
43	12:07:19.8591..	137.135.83.217	192.168.1.109	MQTT	74	Subscribe Request (id=2) [user/response]
47	12:07:19.9805..	137.135.83.217	192.168.1.109	MQTT	68	Subscribe Ack (id=1)
133	12:07:33.9573..	192.168.1.109	137.135.83.217	MQTT	68	Subscribe Ack (id=2)
135	12:07:34.2148..	137.135.83.217	192.168.1.109	MQTT	78	Connect Command
149	12:07:54.1806..	192.168.1.109	137.135.83.217	MQTT	68	Connect Ack
151	12:07:54.2182..	192.168.1.109	137.135.83.217	MQTT	83	Publish Message [water_level/main tank]
153	12:07:54.3403..	137.135.83.217	192.168.1.109	MQTT	115	Publish Message [water_level/house_tank], Publish Message [water_quality/turbidity]
155	12:07:54.4626..	137.135.83.217	192.168.1.109	MQTT	83	Publish Message [water_level/main tank]
164	12:08:05.5578..	192.168.1.109	137.135.83.217	MQTT	115	Publish Message [water_level/house_tank], Publish Message [water_quality/turbidity]
166	12:08:05.7185..	137.135.83.217	192.168.1.109	MQTT	56	Ping Request
223	12:08:19.6034..	192.168.1.109	137.135.83.217	MQTT	68	Ping Response
225	12:08:19.7618..	137.135.83.217	192.168.1.109	MQTT	56	Ping Request
327	12:08:36.0144..	192.168.1.109	137.135.83.217	MQTT	68	Ping Response
329	12:08:36.1725..	137.135.83.217	192.168.1.109	MQTT	85	Publish Message [user/request]
330	12:08:36.1763..	192.168.1.109	137.135.83.217	MQTT	85	Publish Message [user/request]
332	12:08:36.3373..	137.135.83.217	192.168.1.109	MQTT	188	Publish Message [user/response]
				MQTT	188	Publish Message [user/response]

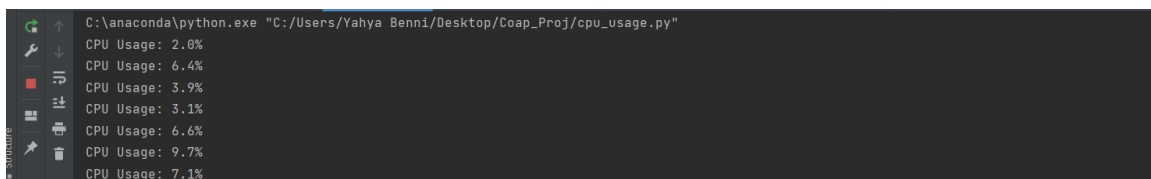
Latency = 54.3403-54.1806= 0.1597 s

Message size: 60,74,83,...Byte

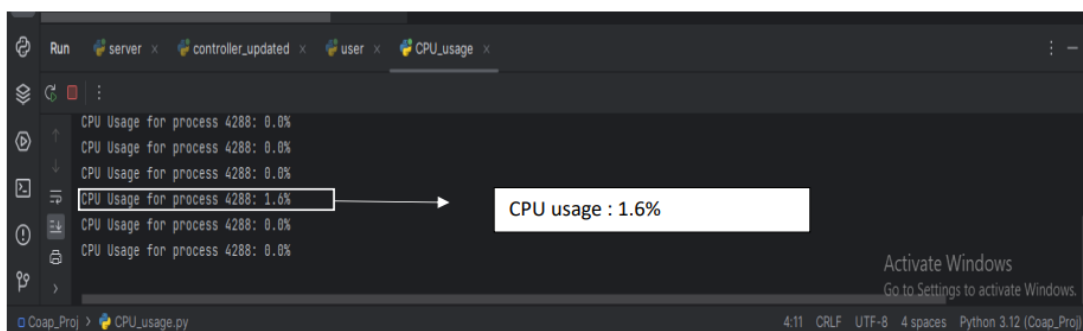
Coap



MQTT CPU usage (between 2.0% and 9.7%)



Coap CPU usage



- **CoAP is more efficient** for sensor data requests due to **lower CPU usage and latency**.
- **MQTT is better for continuous monitoring** since it can provide real-time updates without needing repeated requests.
- **Message sizes in MQTT are larger** due to additional headers, but this is acceptable for non-constrained networks.